

Corso di Robotica

-

Prof.ssa Cecilia Laschi

26/04/2007

Introduzione ai Microcontroller

Dr. Virgilio Mattoli
(mattoli@crim.sssup.it)

Processori Embedded

- ✓ I microprocessori embedded sono contenuti in tutto ciò che ci circonda.
- ✓ I primi microprocessori sono apparsi negli anni '70 → Intel 8080
- ✓ Oggi sono integrati praticamente in ogni apparecchio elettronico presente sulle mercato: lavatrici, forni a microonde, telefonini, autovetture, ...
- ✓ Ogni anno vengono venduti nel mondo **miliardi** di processori embedded

Processori Embedded

Mediamente un microprocessore per essere definito embedded deve avere le seguenti caratteristiche:

→ Deve essere dedicato al controllo real-time di uno specifico dispositivo o funzione.

→ Deve contenere il proprio programma operativo in qualche tipo di memoria non volatile

→ Deve essere trasparente all'utente (deve funzionare come un hardware dedicato)

Processori Embedded

Un *sistema embedded* deve contenere solitamente le seguenti componenti:

- ✓ Un microprocessore
- ✓ Memoria RAM (random access memory)
- ✓ Memoria non-volatile : ROM, EEPROM,, FLASH, ...
- ✓ I/O (interfaccia con l'ambiente)

In cosa un sistema embedded è diverso da un computer (PC)? **Risorse**

	PC	Embedded System
RAM	GB	Centinaia di GByte
ROM	Centinaia di GByte	KByte.

Processori Embedded

In cosa un sistema embedded è diverso da un computer (PC)? **Applicazione!**

→ Personal Computer devono poter svolgere una varietà virtualmente illimitata di funzioni e programmi e applicazioni;

→ I sistemi embedded devono svolgere un limitato numero di task (as. controllo della temperatura di un forno, controllo del tempo di lavaggio di una lavatrice,)

Processori Embedded

Perche usare un sistema a microprocessore embedded?

Costo. – Microprocessore embedded è molto vantaggioso rispetto all'implementazione hardware in componenti discreti

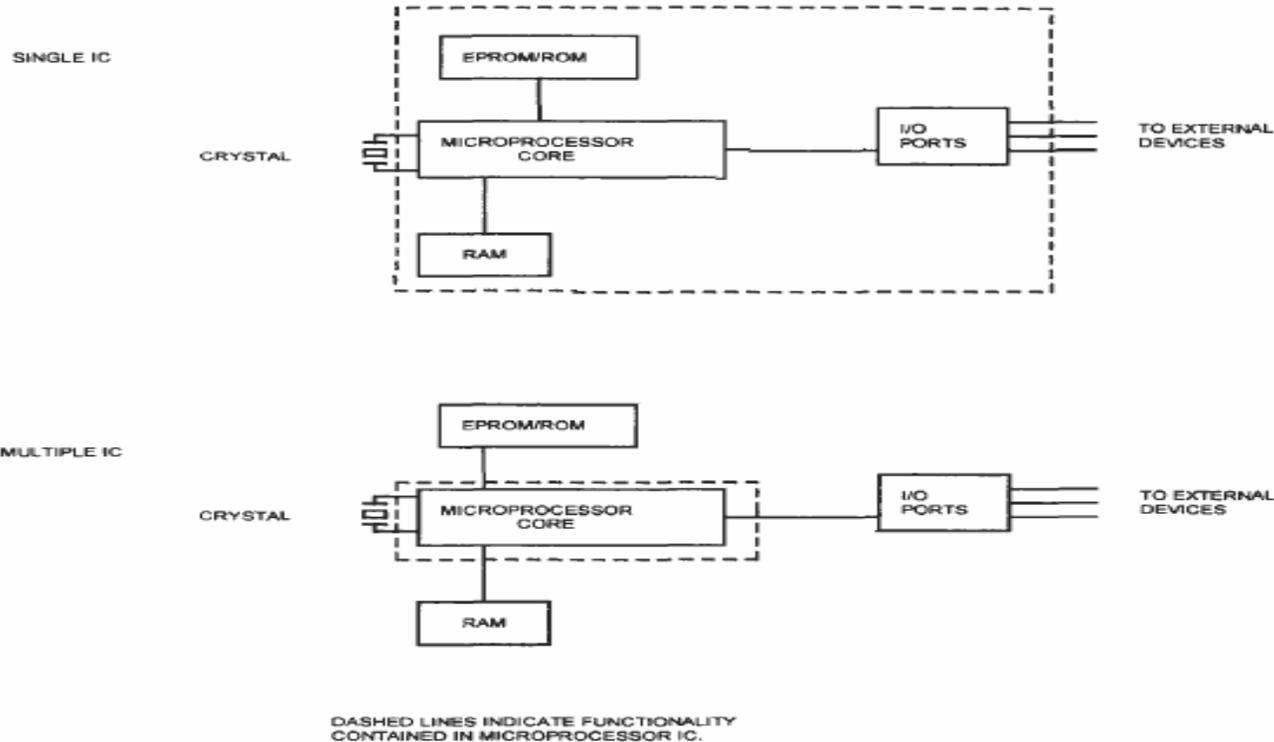
Programmabilità – La stessa piattaforma hardware permette di implementare differenti applicazioni

Flessibilità - Le funzionalità possono essere semplicemente ri-programmate in firmware

Adattabilità – Implementazione di sistemi intelligenti (“smart”) con capacità di adattarsi all'ambiente

Cosa è un microcontroller?

Microcontroller = Microprocessore Embedded in un singolo chip

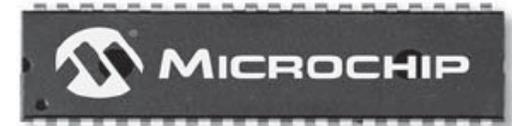


Un microcontroller è progettato per minimizzare il numero dei componenti richiesti per la realizzazione di un sistema embedded, incorporando memoria e I/O.

Spesso sono specializzati per una certa applicazione (a scapito della flessibilità).

Cosa è un microcontroller?

Dimensioni e Packaging



P: 40-pin PDIP
(52.27 x 15.24 x 3.81 mm)



PF: 64-pin TQFP
(14 x 14 x 1 mm)



PT: 64-pin TQFP
(10 x 10 x 1 mm)



PF: 80-pin TQFP
(14 x 14 x 1 mm)



PF: 100-pin TQFP
(14 x 14 x 1 mm)



SO: 18-pin SOIC
(11.53 x 10.34 x 2.31 mm)



P: 18-pin PDIP
(22.81 x 7.95 x 3.3 mm)



S: 20-pin SSOP
(2 x 7.85 x 1.85 mm)



SO: 20-pin SOIC
(12.80 x 10.34 x 2.31 mm)



PT: 44-pin TQFP
(10 x 10 x 1 mm)



PT: 80-pin TQFP
(12 x 12 x 1 mm)



PT: 100-pin TQFP
(12 x 12 x 1 mm)



P: 20-pin PDIP
(26.24 x 7.87 x 3.3 mm)



ML: 28-pin QFN
(6 x 6 x 0.9 mm)



SO: 28-pin SOIC
(17.88 x 10.34 x 2.31 mm)



ML: 44-pin QFN
(8 x 8 x 0.9 mm)



MM: 28-pin QFN
(6 x 6 x 0.9 mm)



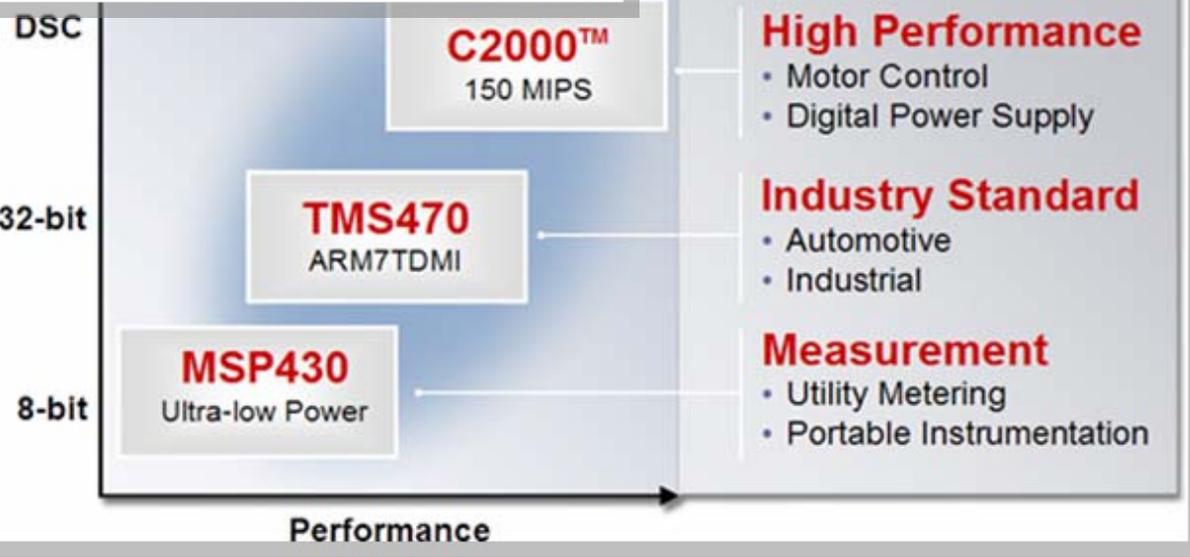
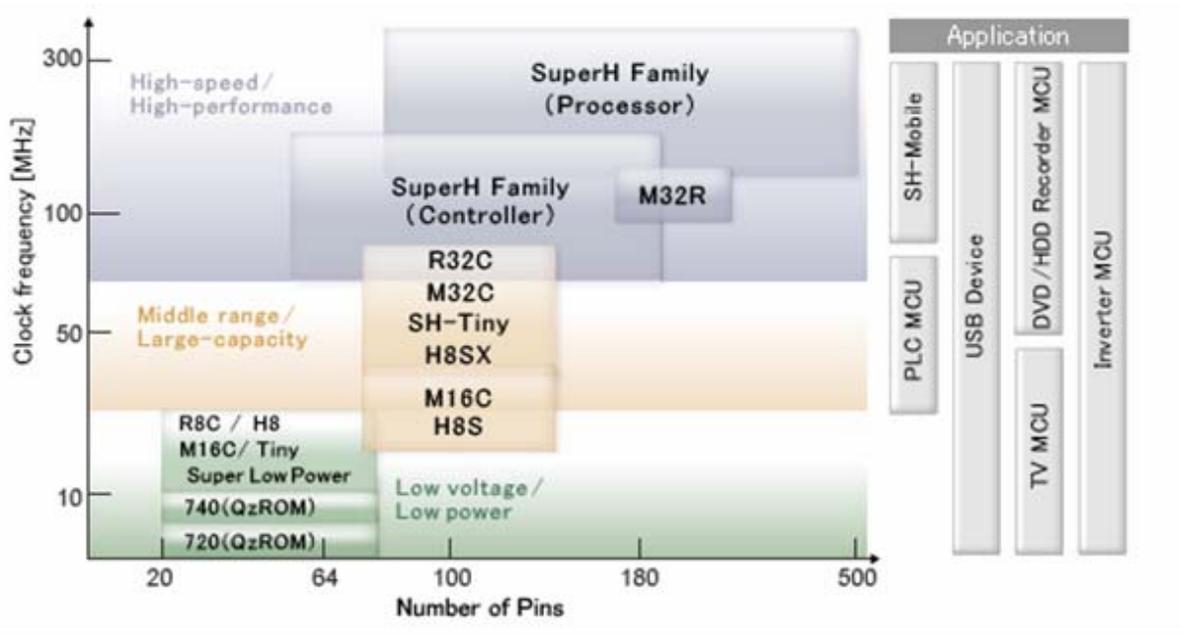
SP: 28-pin SPDIP
(34.67 x 7.87 x 3.3 mm)



SS: 28-pin SSOP

Cosa è un microcontroller?

8, 16 o 32 bit



Cosa è un microcontroller?

Principali Produttori



<http://www.microchip.com>



<http://www.ti.com/>



<http://www.atmel.com/>

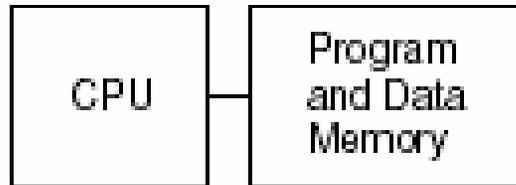


<http://eu.renesas.com/>

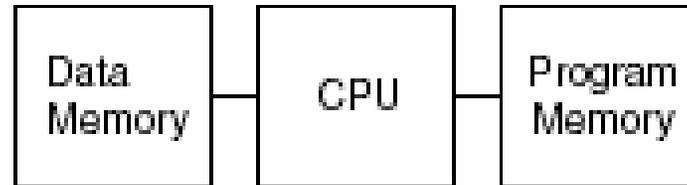
Architettura di un microcontroller

I sistemi a microprocessore hanno solitamente un'architettura di tipo von Neumann con una singola memoria per il programma e i dati che permette la massima flessibilità di allocazione; i microcontroller hanno invece tipicamente un'architettura di tipo Harvard in cui la memoria di programma è separata da quella per i dati.

von Neumann architecture



Harvard architecture



Il vantaggio dell'architettura Harvard per applicazioni embedded è dovuta alla possibilità di usare due tipi diversi di memoria per i dati e il programma

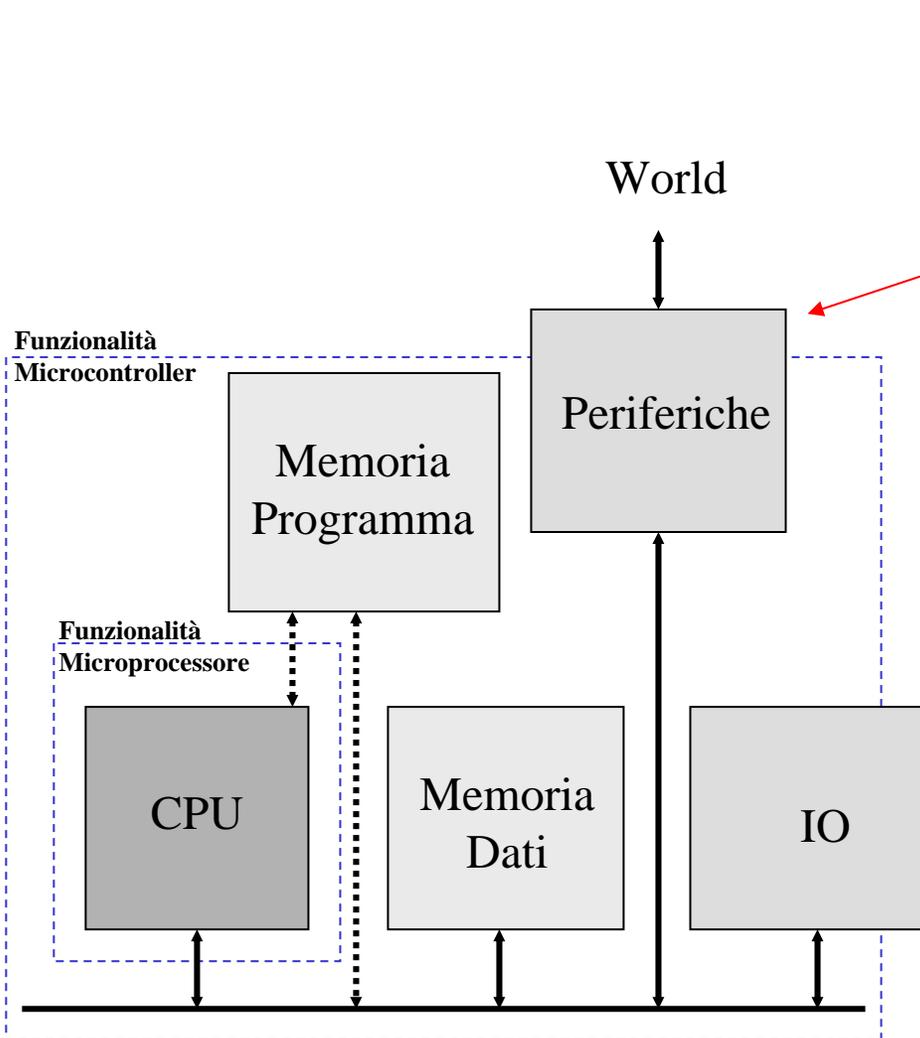
Programma → memoria non-volatile (ROM, programma non si perde allo spegnimento)

Dati variabili → RAM volatile

Un altro potenziale vantaggio dell'architettura Harvard è dato dal fatto che il trasferimento dei dati e delle istruzioni di programma avviene in parallelo (velocità doppia).

Architettura generica di un microcontroller

Tipica architettura di un microcontroller: CPU, memorie, I/O e periferiche per l'interfaccia con l'esterno. Componenti collegate da un Bus comune!



Periferiche: timers, counters, porte scambio dati (seriali parallele) convertitori Analogico-Digitale (DAC) e Digitale-Analogico (DAC) integrati.

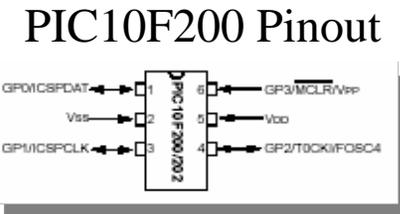
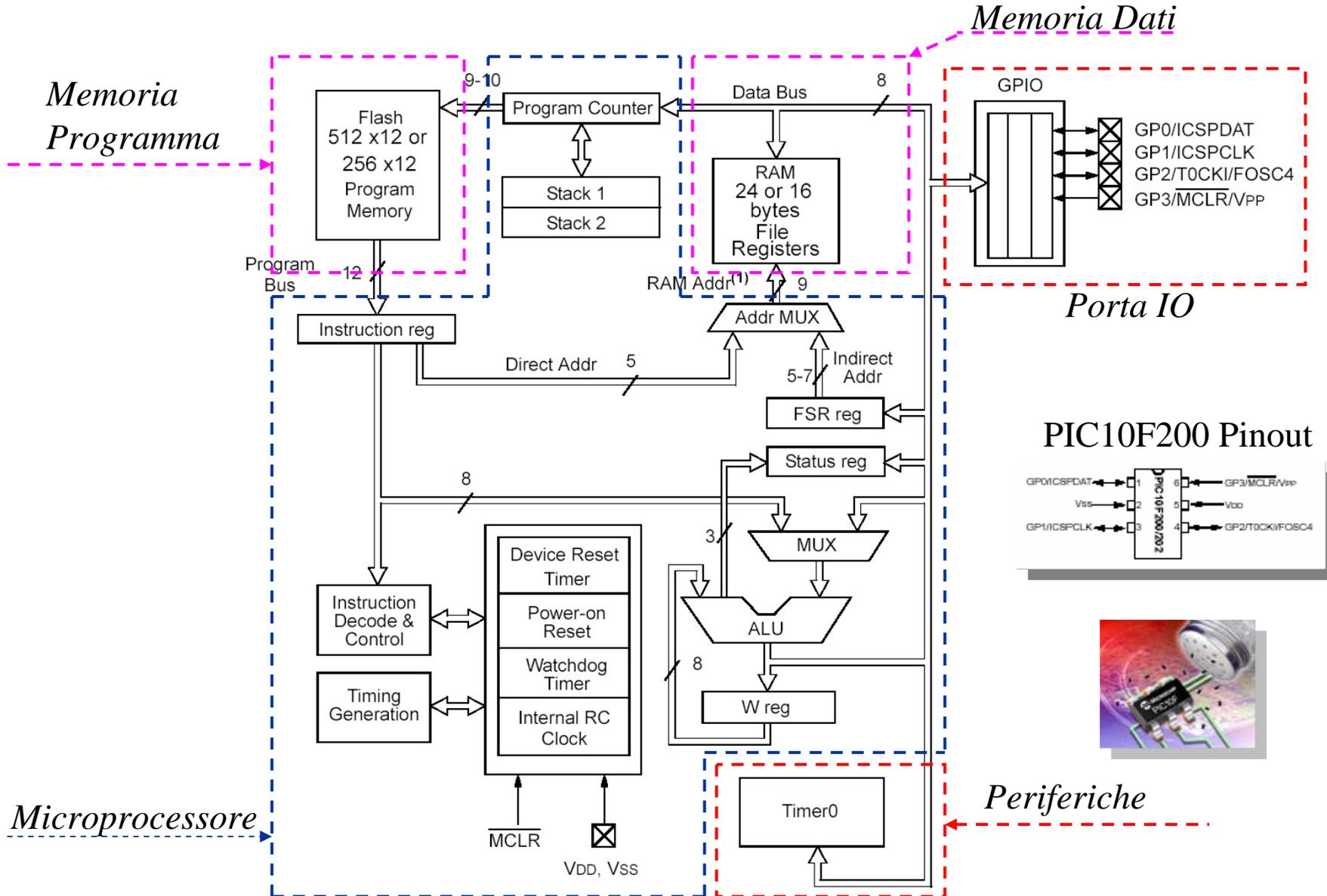
Vantaggi dell'integrazione

- Molteplici funzionalità in chip singolo
- Costi e dimensioni minori
- Minor consumo energetico
- Minor numero di connessioni esterne
- Più pin disponibili per I/O
- Maggior affidabilità del sistema (minor numero di componenti)

Svantaggi dell'integrazione

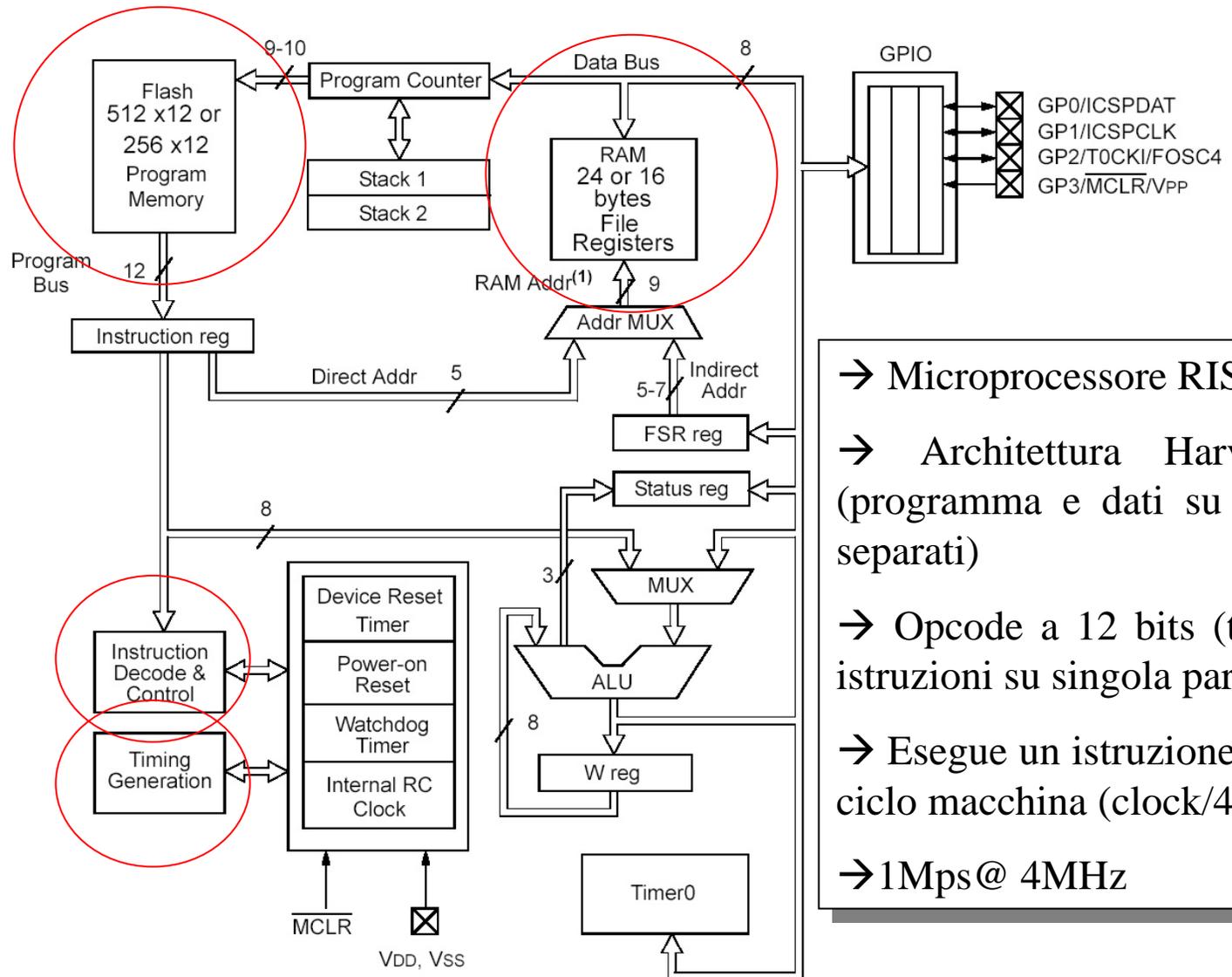
- Flessibilità delle periferiche ridotta
- Limitata espandibilità (memoria e IO)
- Performance minori di periferiche e IO

Architettura di un semplice microcontrollore - PIC10F200 (Microchip)



Architettura di un semplice microcontrollore - PIC10F200

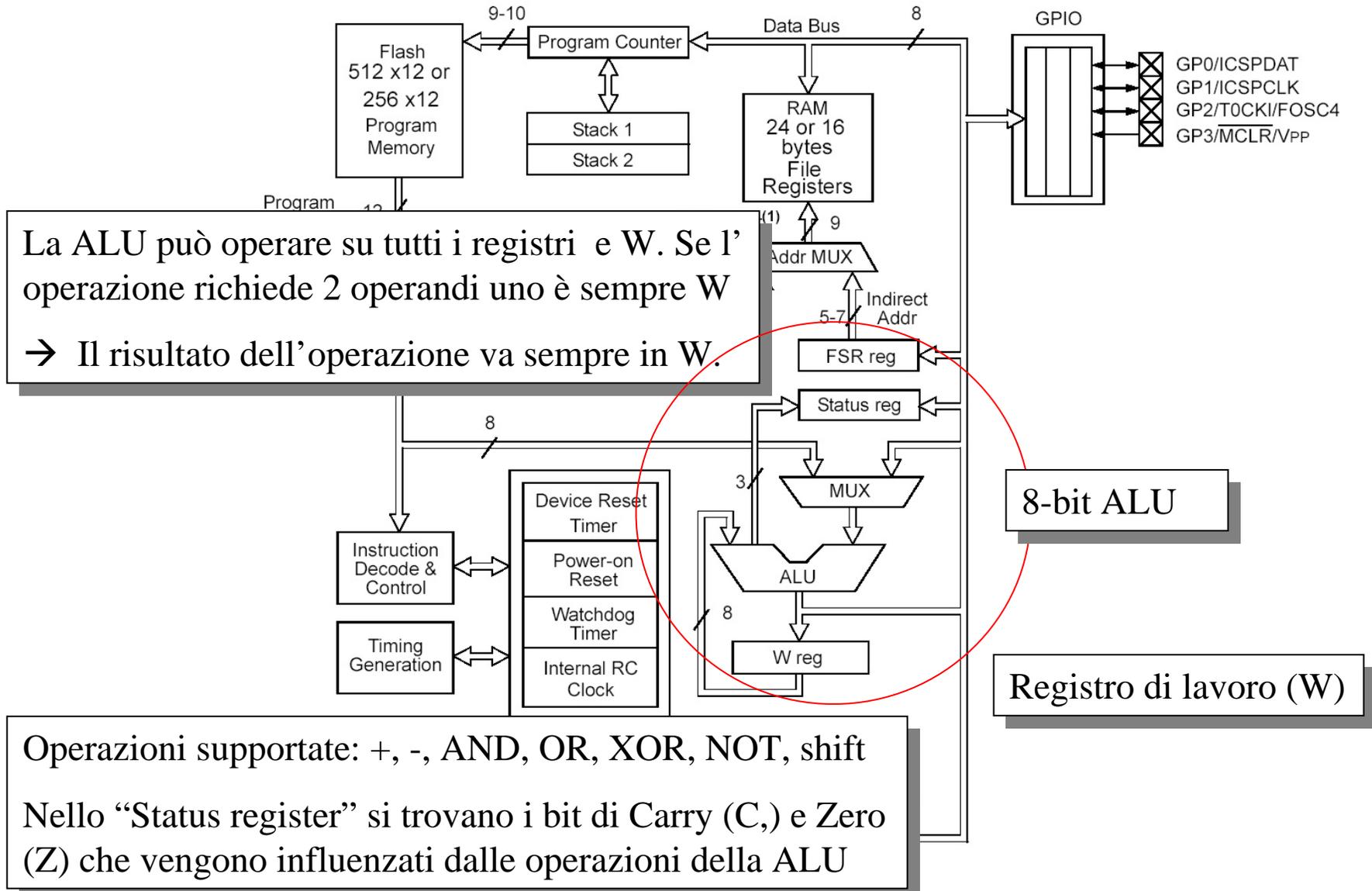
CPU



- Microprocessore RISC
- Architettura Harvard (programma e dati su bus separati)
- Opcode a 12 bits (tutte istruzioni su singola parola)
- Esegue un'istruzione per ciclo macchina (clock/4)
- 1Mps@ 4MHz

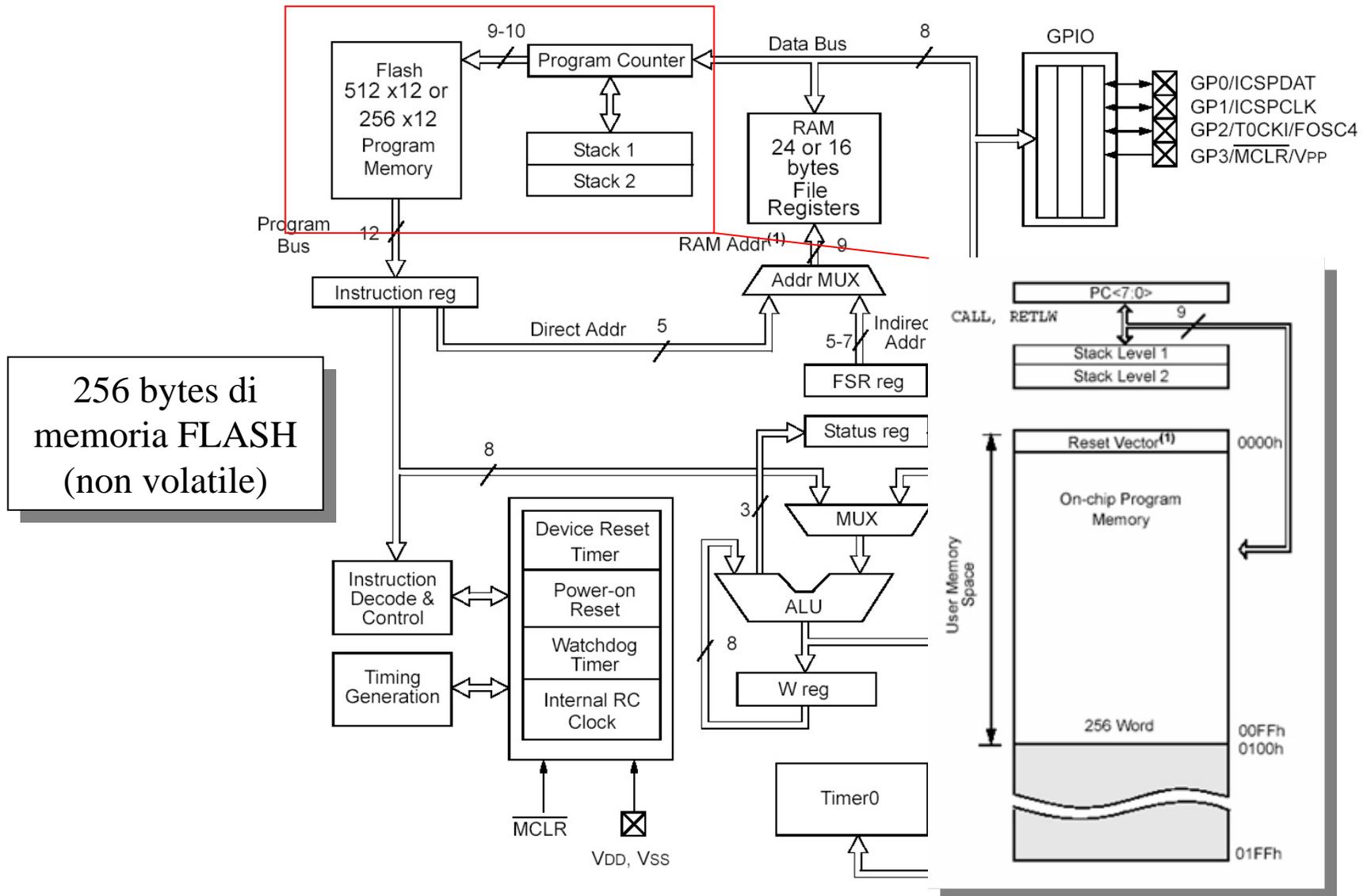
Architettura di un semplice microcontrollore - PIC10F200

CPU



Architettura di un semplice microcontrollore - PIC10F200

Memoria di Programma

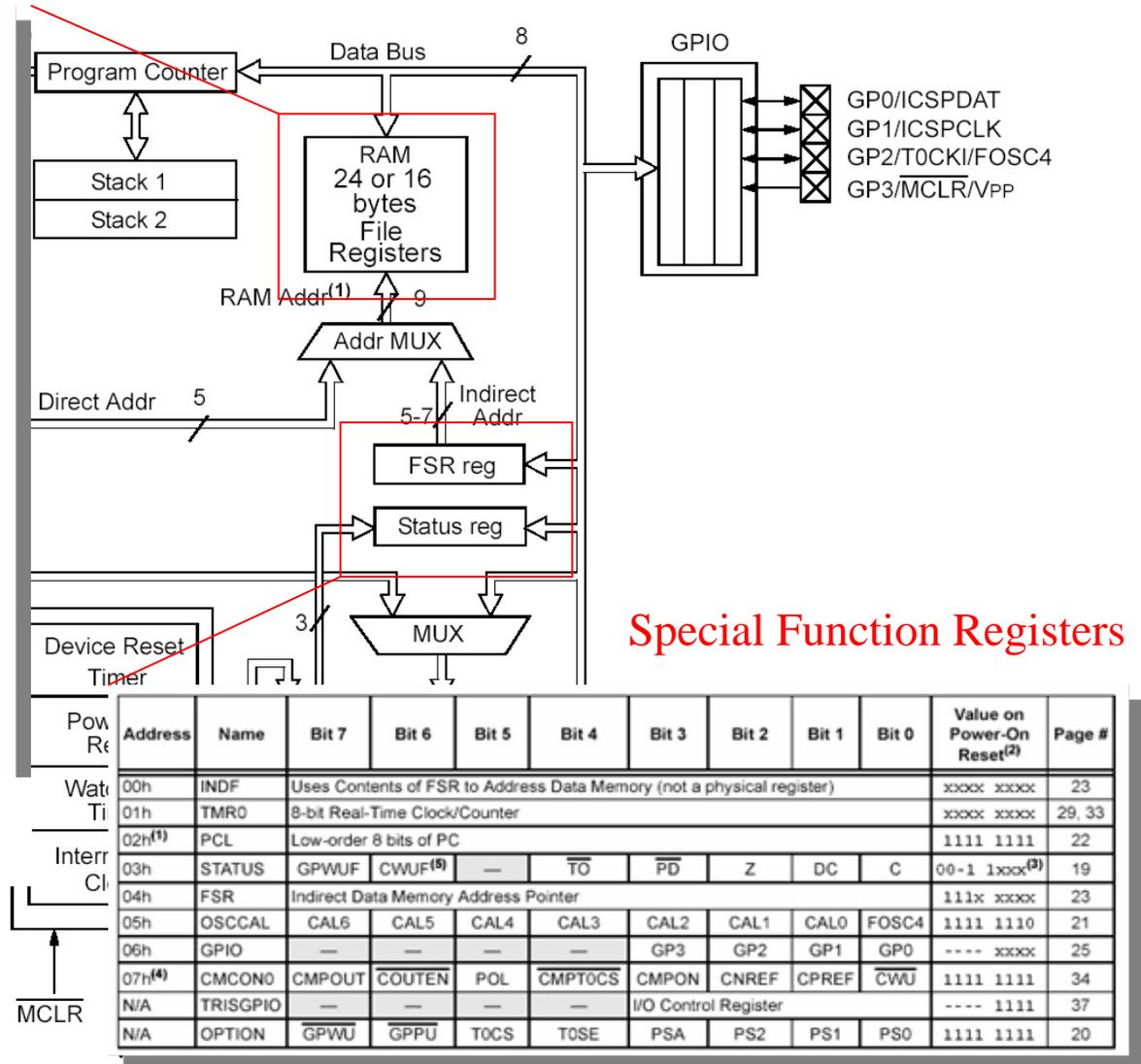


Architettura di un semplice microcontrollore - PIC10F200

Memoria Dati

File Address

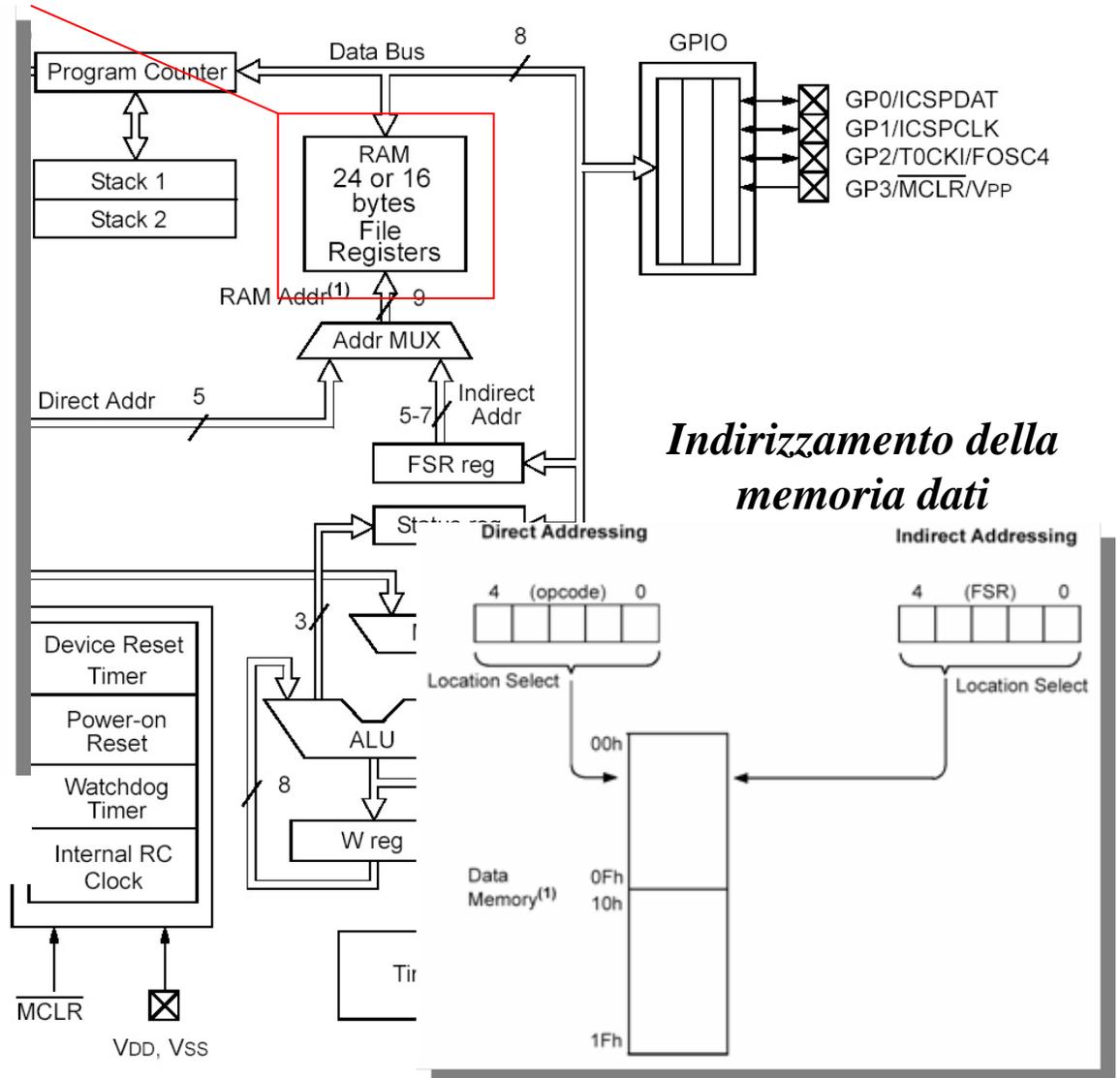
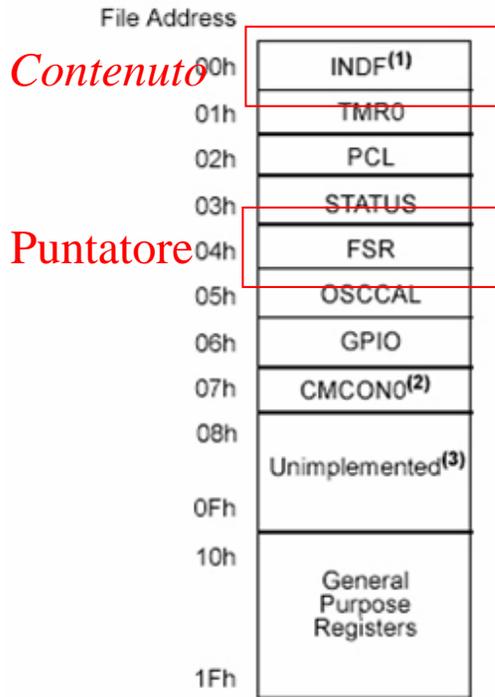
00h	INDF ⁽¹⁾
01h	TMR0
02h	PCL
03h	STATUS
04h	FSR
05h	OSCCAL
06h	GPIO
07h	CMCON0 ⁽²⁾
08h	Unimplemented ⁽³⁾
0Fh	
10h	General Purpose Registers
1Fh	



16 bytes di
memoria RAM
statica
(volatile)

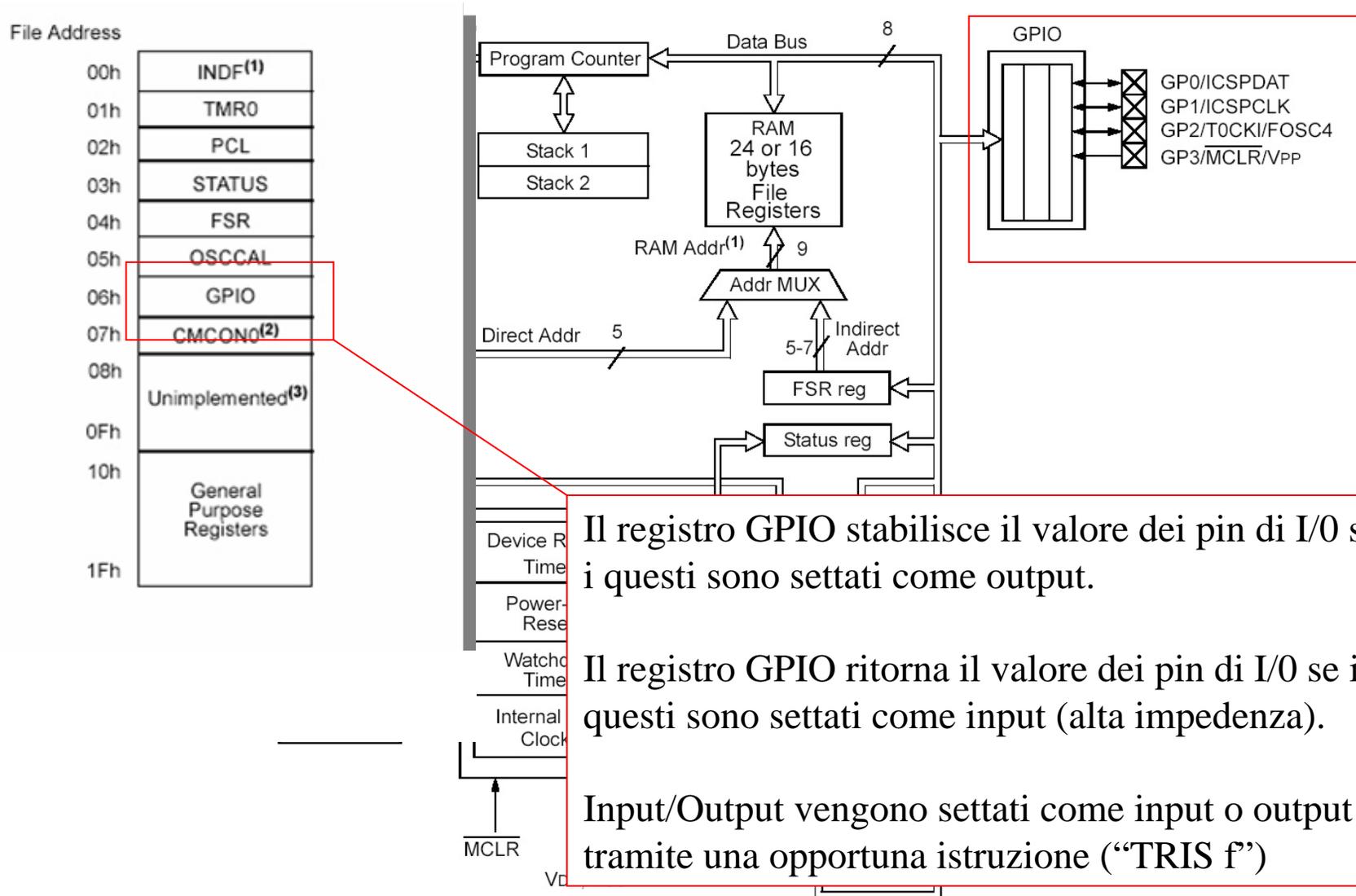
Architettura di un semplice microcontrollore - PIC10F200

Memoria Dati



Architettura di un semplice microcontrollore - PIC10F200

Pin di Input/Output (Porta bidirezionale)



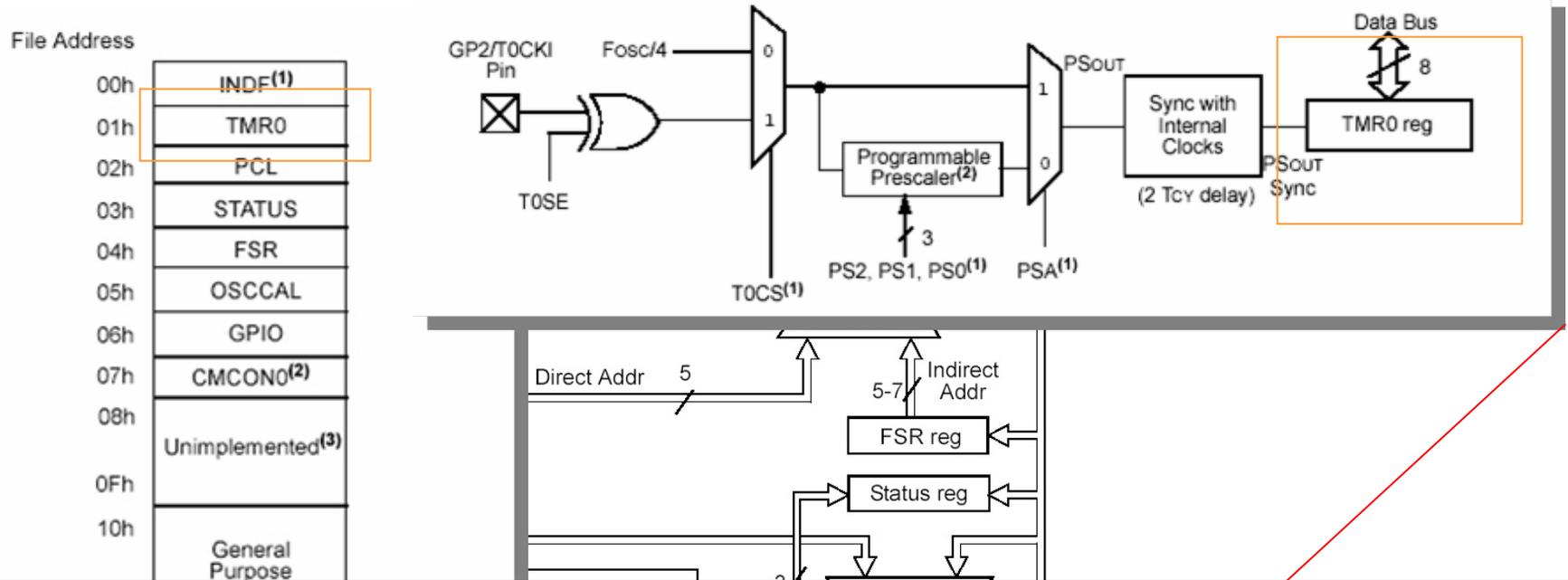
Il registro GPIO stabilisce il valore dei pin di I/O se i questi sono settati come output.

Il registro GPIO ritorna il valore dei pin di I/O se i questi sono settati come input (alta impedenza).

Input/Output vengono settati come input o output tramite una opportuna istruzione (“TRIS f”)

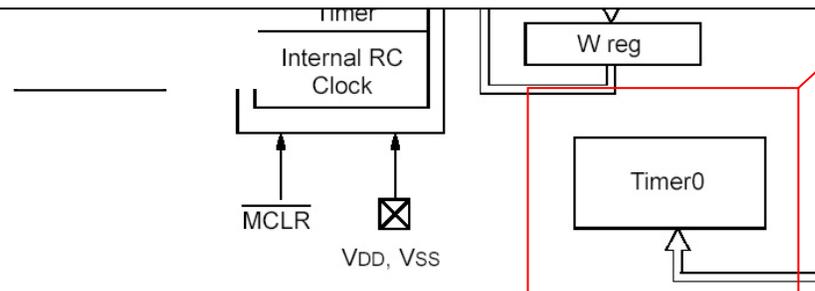
Architettura di un semplice microcontrollore - PIC10F200

Pin di Input/Output



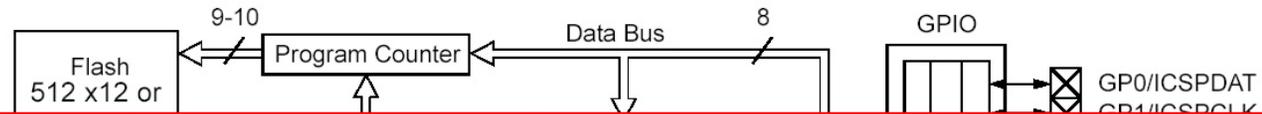
→ Il Timer è una parte cruciale per ogni sistema embedded. In questo caso il timer consiste in un semplice contatore a 8 bit che può essere letto e scritto da programma.

→ Il Timer può essere incrementato da impulsi esterni o internamente dal ciclo di sistema



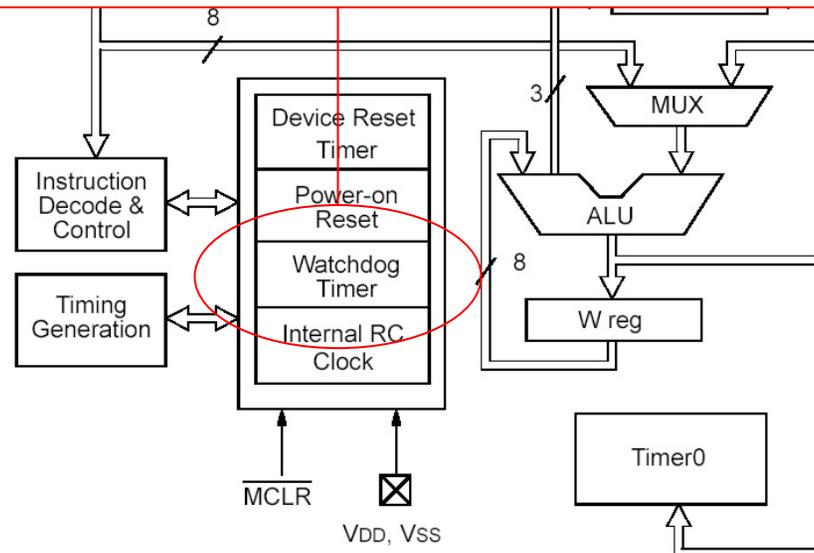
Architettura di un semplice microcontrollore - PIC10F200

Wathcdog Timer



Il Watchdog WDT è un contatore basato su un oscillatore indipendente che deve essere continuamente azzerato da programma. Se il WDT va in overflow effettua un reset del microcontroller.

Questo sistema di sicurezza impedisce che bugs del programma o eventi imprevisti (es. una scarica elettrostatica) possano mandare il microcontroller in uno stato incongruente con il programma.



Architettura di un semplice microcontrollore - PIC10F200

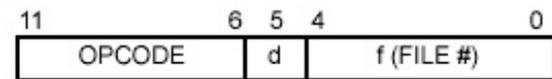
Il set di istruzioni

Il set di istruzioni per il PIC10F200 è composto da 33 istruzioni divise in tre categorie:

- Operazioni Byte-oriented
- Operazioni Bit-oriented
- Operazioni con costanti e di controllo

Ogni istruzione è una parola **12-bit** che contiene l'**opcode** (che specifica il tipo di istruzione) e uno o più **operandi**

Byte-oriented file register operations

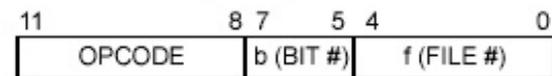


d = 0 for destination W

d = 1 for destination f

f = 5-bit file register address

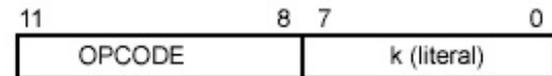
Bit-oriented file register operations



b = 3-bit address

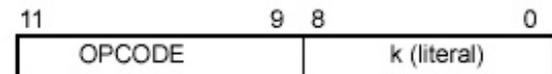
f = 5-bit file register address

Literal and control operations (except GOTO)



k = 8-bit immediate value

Literal and control operations – GOTO instruction



k = 9-bit immediate value

Architettura di un semplice microcontrollore - PIC10F200

Il set di istruzioni

Byte-oriented

Mnemonic, Operands	Description	Cycles	12-Bit Opcode			Status Affected	Notes
			MSb	LSb			
ADDWF f, d	Add W and f	1	0001	11dF	ffff	C, DC, Z	1, 2, 4
ANDWF f, d	AND W with f	1	0001	01dF	ffff	Z	2, 4
CLRF f	Clear f	1	0000	011f	ffff	Z	4
CLRW -	Clear W	1	0000	0100	0000	Z	
COMF f, d	Complement f	1	0010	01dF	ffff	Z	
DECF f, d	Decrement f	1	0000	11dF	ffff	Z	2, 4
DECFSZ f, d	Decrement f, Skip if 0	1(2)	0010	11dF	ffff	None	2, 4
INCF f, d	Increment f	1	0010	10dF	ffff	Z	2, 4
INCFSZ f, d	Increment f, Skip if 0	1(2)	0011	11dF	ffff	None	2, 4
IORWF f, d	Inclusive OR W with f	1	0001	00dF	ffff	Z	2, 4
MOVF f, d	Move f	1	0010	00dF	ffff	Z	2, 4
MOVWF f	Move W to f	1	0000	001F	ffff	None	1, 4
NOP -	No Operation	1	0000	0000	0000	None	
RLF f, d	Rotate left f through Carry	1	0011	01dF	ffff	C	2, 4
RRF f, d	Rotate right f through Carry	1	0011	00dF	ffff	C	2, 4
SUBWF f, d	Subtract W from f	1	0000	10dF	ffff	C, DC, Z	1, 2, 4
SWAPF f, d	Swap f	1	0011	10dF	ffff	None	2, 4
XORWF f, d	Exclusive OR W with f	1	0001	10dF	ffff	Z	2, 4

Bit-oriented

BCF f, b	Bit Clear f	1	0100	bbbF	ffff	None	2, 4
BSF f, b	Bit Set f	1	0101	bbbF	ffff	None	2, 4
BTFSZ f, b	Bit Test f, Skip if Clear	1(2)	0110	bbbF	ffff	None	
BTFSZ f, b	Bit Test f, Skip if Set	1(2)	0111	bbbF	ffff	None	

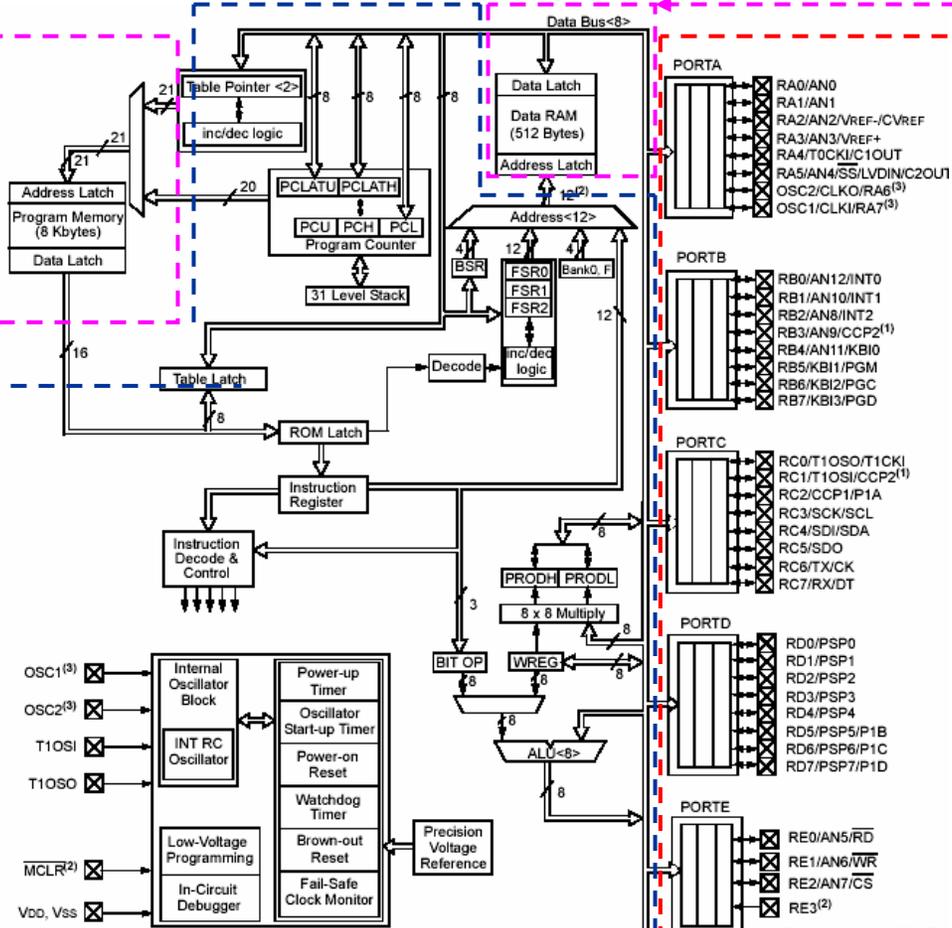
Costanti e controllo

ANDLW k	AND literal with W	1	1110	kkkk	kkkk	Z	
CALL k	Call Subroutine	2	1001	kkkk	kkkk	None	1
CLRWDT	Clear Watchdog Timer	1	0000	0000	0100	<u>TO, PD</u>	
GOTO k	Unconditional branch	2	101k	kkkk	kkkk	None	
IORLW k	Inclusive OR literal with W	1	1101	kkkk	kkkk	Z	
MOVLW k	Move literal to W	1	1100	kkkk	kkkk	None	
OPTION -	Load Option register	1	0000	0000	0010	None	
RETLW k	Return, place Literal in W	2	1000	kkkk	kkkk	None	
SLEEP -	Go into Standby mode	1	0000	0000	0011	<u>TO, PD</u>	
TRIS f	Load TRIS register	1	0000	0000	0fff	None	3
XORLW k	Exclusive OR literal to W	1	1111	kkkk	kkkk	Z	

Architettura di un microcontrollore più complesso - PIC18F4320 (Microchip)

Memoria Programma

Memoria Dati



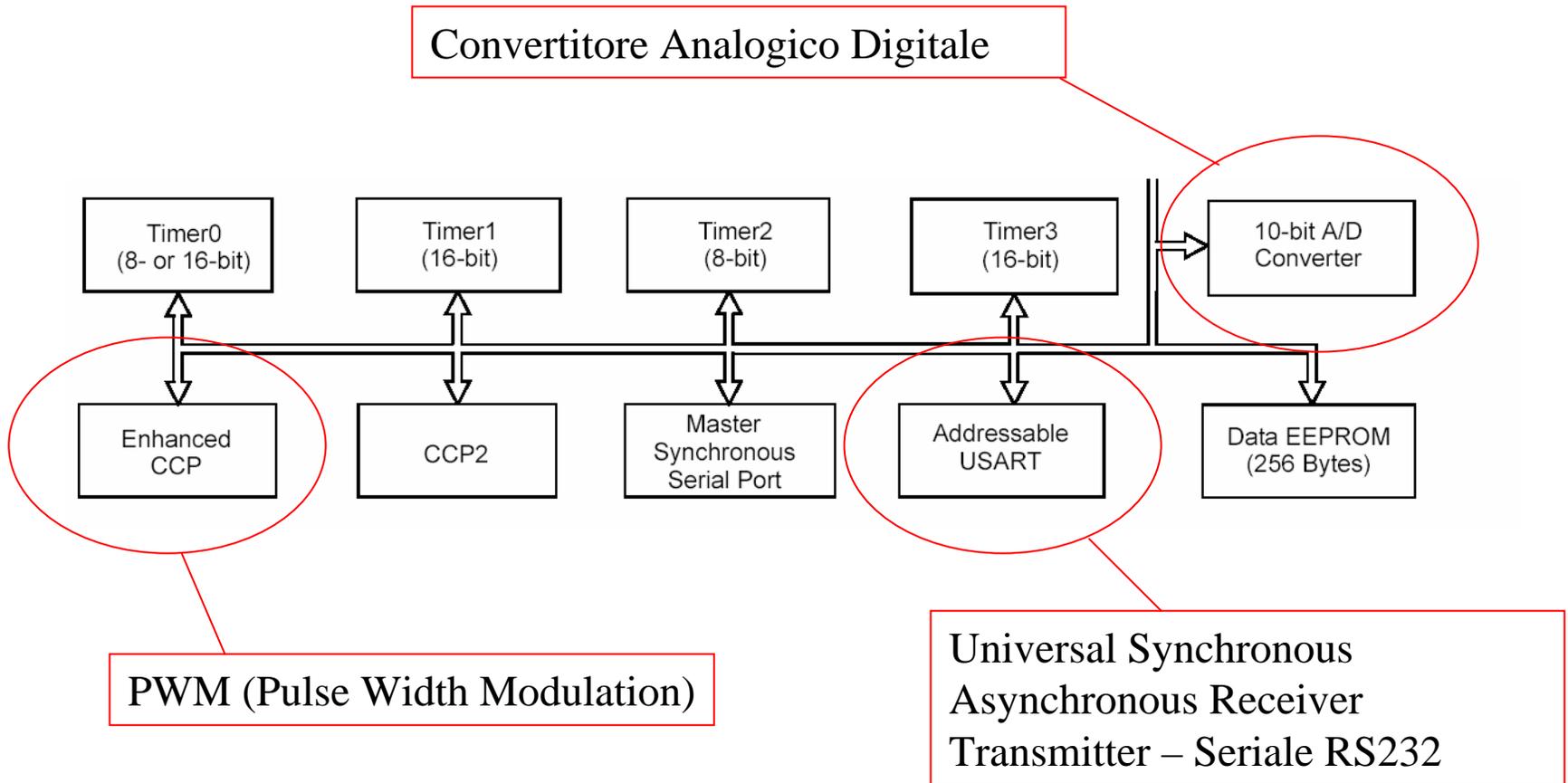
Microprocessore

Porte IO

Periferiche

Architettura di un microcontrollore più complesso - PIC18F4320 (Microchip)

Altre Periferiche



Architecture Software

Il software per sistemi embedded è di solito implementato sulla base di una delle seguenti architetture:

Singolo ciclo di polling - Singolo pezzo di codice che testa continuamente un certo input ed esegue di conseguenza le routines necessarie

Macchina a stati - Compartimentazione delle funzionalità, richiede duplicazione del codice, adatto a applicazioni che effettuano singole funzioni

Macchina a stati multipla/Ciclo di polling - un ciclo di polling per ogni processo, ogni ciclo di polling esegue le routines relative allo stato attuale, finito un ciclo si passa al successivo

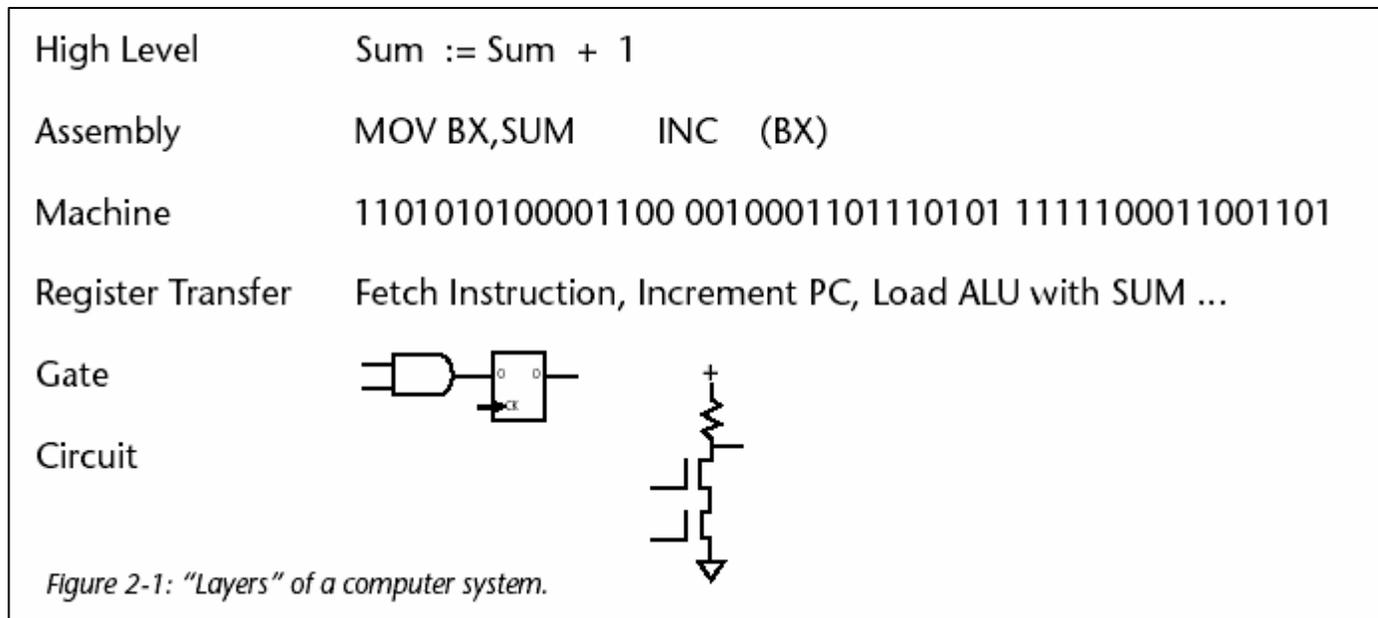
RTOS - Sistema Operativo Real-time : permette l'esecuzione di più task in parallelo

The Development Language

Linguaggio di alto livello vs. Assembler

- Permette applicazioni più complesse
- Riutilizzabilità
- Portabilità
- Richiede maggiori risorse

- Velocità
- Miglior controllo
- Strumenti di sviluppo gratuiti
- Specifico per il processore



La scelta del microcontroller

Solitamente più scelte diverse sono possibili: nello specifico vanno tenuti in considerazione i seguenti parametri sulla base dell'applicazione finale.

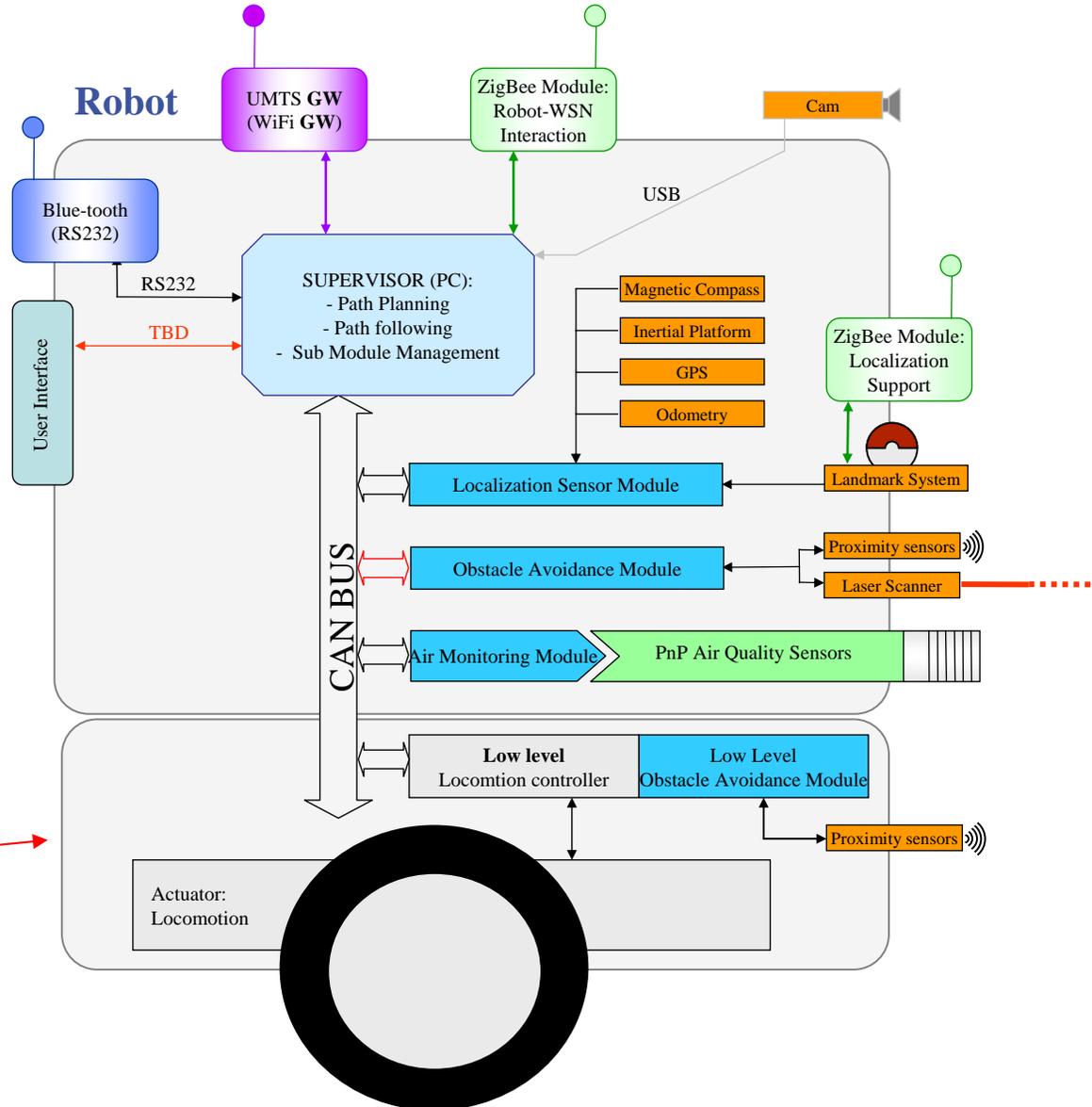
- ✓ Numero dei pin di I/O
- ✓ Interfacce
- ✓ Memoria RAM (quantità e tipo)
- ✓ Numero e tipo di interrupts
- ✓ Velocità del processore
- ✓ Consumi energetici
- ✓ Memoria di programma (quantità e tipo)
- ✓ Ambiente di sviluppo

Microprocessori in robotica

- ✓ Controllo dei motori/attuatori (basso livello)
- ✓ Controllo movimento (alto livello)
- ✓ Power management (carica/scarica batterie)
- ✓ Acquisizione/elaborazione dei segnali dei sensori (prossimità, encoders, odometria, contatto, visione)
- ✓ Interfacce

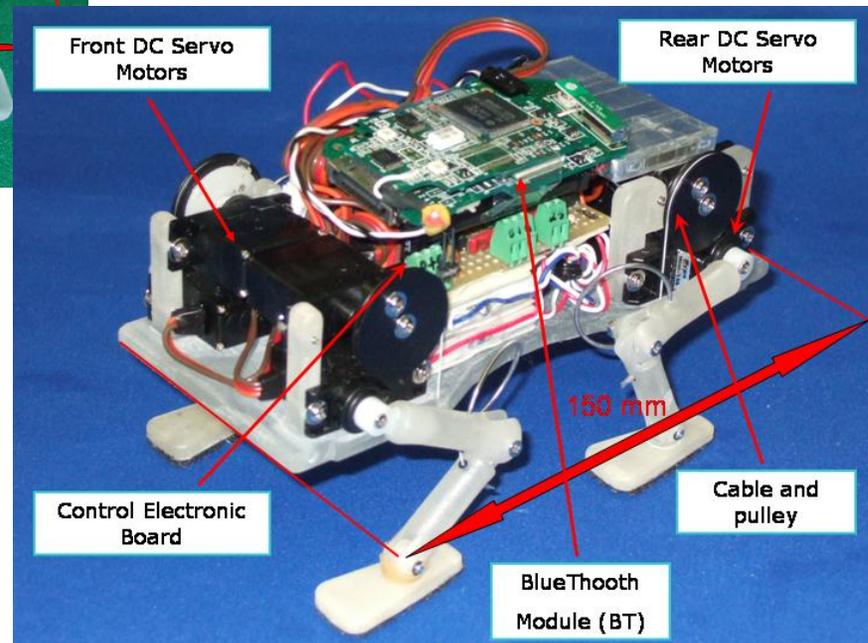
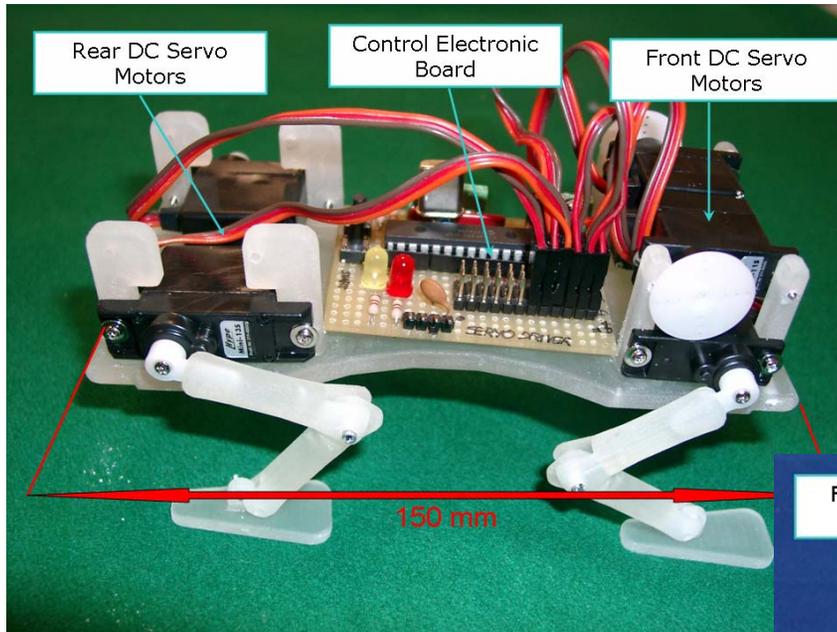
Microprocessori in robotica

Architettura di un sistema robotico - Esempio 1: DustBot



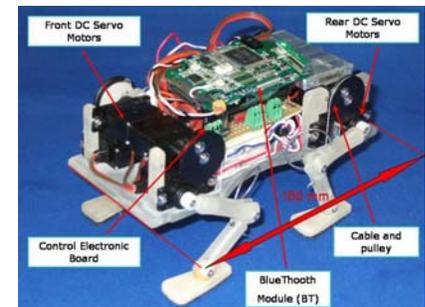
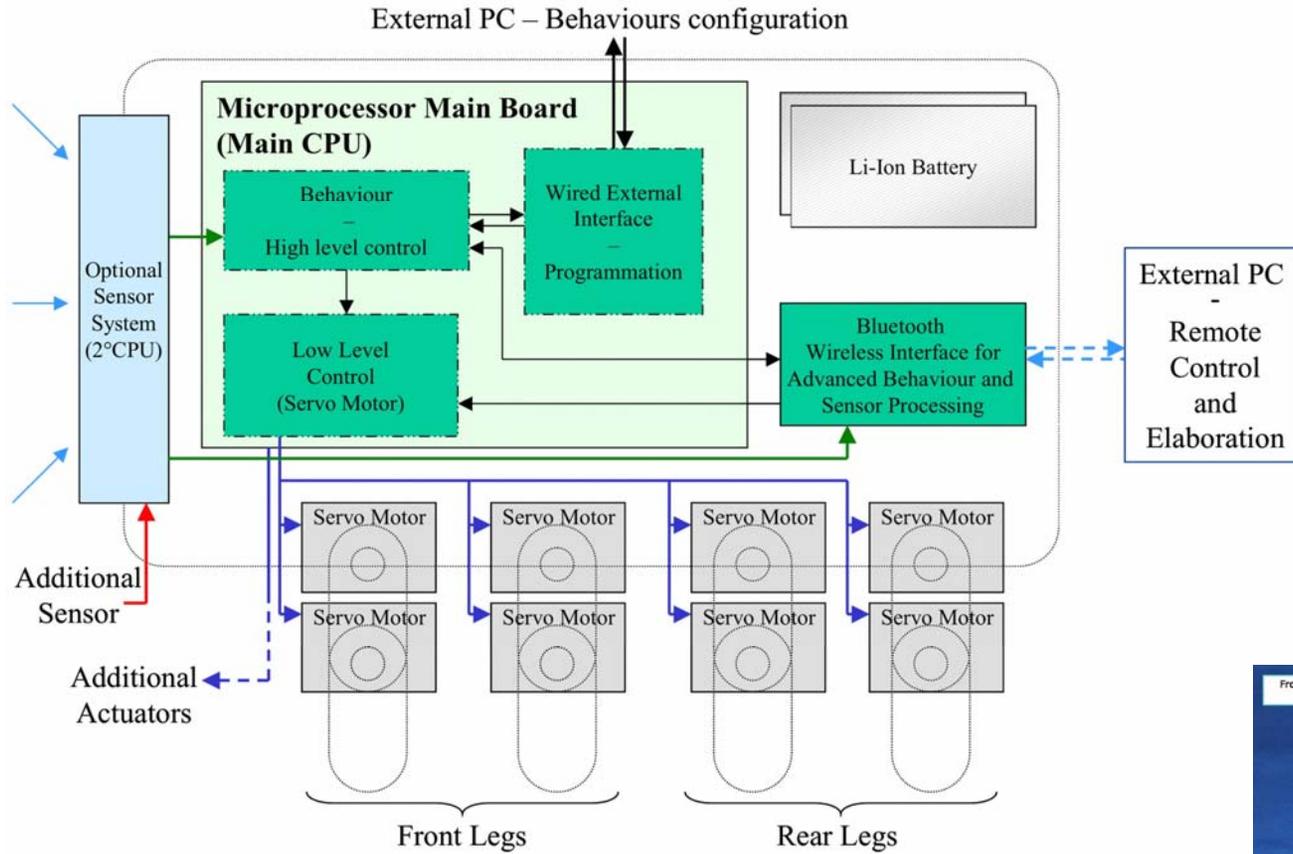
Microprocessori in robotica

Architettura di un sistema robotico - Esempio 2: Robot biomimetico a 4 zampe



Microprocessori in robotica

Architettura di un sistema robotico - Esempio 2: Robot biomimetico a 4 zampe



Microprocessori in robotica

Architettura di un sistema robotico - Esempio 2: Robot biomimetico a 4 zampe

